

## 자료구조 5주차 과제



## 1. Lab3 ap1.c

```
#include <stdio.h>
#include <stdlib.h>
void main() {
    int list[5]; //5 개짜리 배열을 만듦
    int *plist[5] = {NULL,}; //5 개의 포인터 배열을 만듦
    plist[0] = (int *)malloc(sizeof(int)); // plist 0 번째 배열에 int 메모리 할당
    list[0] = 1; // list 0 번째 자리에 1 할당
    list[1] = 100; // list 1 번째 자리에 100 할당
    *plist[0] = 200; // 포인터 plist 가 가르키는 곳에 200 할당
    printf("----- 오재식 2022040014 -----");
    printf("list[0] = %d\n", list[0]); // list 0 번째 값 출력
    printf("&list[0] = %p\n", &list[0]); // list 0 번째 주소값 출력
    printf("list = %p\n", list); // list 주소값 출력
    printf("&list = %p\n ", &list); // list 의 주소값 출력
    printf("-----\n\n");
    printf("list[1] = %d\n", list[1]); // list 1 번째 값 출력
    printf("&list[1] = %p\n", &list[1]); // list 1 번째 주소값 출력
    printf("*(list+1) = %d\n", *(list + 1)); // list 1 번째 값 출력
    printf("list+1 = %p\n", list+1); // list 1 번째 주소값 출력

    printf("-----\n\n");
    printf("*plist[0] = %d\n", *plist[0]); // *plist[0]의 할당된 값 출력
    printf("&plist[0] = %p\n", &plist[0]); // plist[0]의 주소값 출력
    printf("&plist = %p\n", &plist); // plist 의 주소값 출력
    printf("plist = %p\n", plist); // plist 주소값 출력
    printf("plist[0] = %p\n", plist[0]); // plist[0]의 할당된 malloc 의 주소값 출력
    printf("plist[1] = %p\n", plist[1]); // 포인터 plist[1]의 할당된 주소값 출력
    printf("plist[2] = %p\n", plist[2]); // 포인터 plist[2]의 할당된 주소값 출력
    printf("plist[3] = %p\n", plist[3]); // 포인터 plist[3]의 할당된 주소값 출력
    printf("plist[4] = %p\n", plist[4]); // 포인터 plist[4]의 할당된 주소값 출력

    free(plist[0]); // plist[0]의 할당된 메모리 해제
}
```

```
----- 오재식 2022040014 -----list[0] = 1
&list[0] = 0x16fb8f2c4
list = 0x16fb8f2c4
&list = 0x16fb8f2c4

list[1] = 100
&list[1] = 0x16fb8f2c8
*(list+1) = 100
list+1 = 0x16fb8f2c8

*plist[0] = 200
&plist[0] = 0x16fb8f298
&plist = 0x16fb8f298
plist = 0x16fb8f298
plist[0] = 0x14be06780
plist[1] = 0x0
plist[2] = 0x0
plist[3] = 0x0
plist[4] = 0x0
```

배열은 0번째 인덱스의 주소값과 배열 이름의 값과

배열의 주소값이 일치함.

## 2. Lab3 ap2.c

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int list[5]; // 5 개짜리 배열을 만듦
    int *plist[5]; // 5 개짜리 포인터 배열을 만듦

    list[0] = 10; // list[0]에 10 할당
    list[1] = 11; // list[1]에 11 할당

    plist[0] = (int*)malloc(sizeof(int)); //plist[0]에 int 메모리 주소 할당
    printf("----- 오재식 2022040014 -----");
    printf("list[0] \t= %d\n", list[0] ); //list[0]에 할당된 값 출력
    printf("list \t\t= %p\n", list); // list의 주소값 출력
    printf("&list[0] \t= %p\n", &list[0]); // list[0]번째 주소값 출력
    printf("list+0 \t= %p\n", list+0); // list의 주소값 출력
    printf("list+1 \t= %p\n", list+1); // list 1 번째 주소값 출력
    printf("list+2 \t= %p\n", list+2); // list 2 번째 주소값 출력
    printf("list+3 \t= %p\n", list+3); // list 3 번째 주소값 출력
    printf("list+4 \t= %p\n", list+4); // list 4 번째 주소값 출력
    printf("&list[4] \t= %p\n", &list[4]); // list[4]번째 주소값 출력

    free(plist[0]); // plist[0]번째에 할당된 메모리값 해제
    return 0;
}
```

```
----- 오재식 2022040014 -----
list[0]      = 10
list         = 0x16f3032c4
&list[0]     = 0x16f3032c4
list+0      = 0x16f3032c4
list+1      = 0x16f3032c8
list+2      = 0x16f3032cc
list+3      = 0x16f3032d0
list+4      = 0x16f3032d4
&list[4]    = 0x16f3032d4
```

배열의 인덱스 값의 시작이 0인 이유는 배열 시작의 첫번째 주소값이 배열 시작 주소의 0을 더한 것이기 때문이다. 배열에 1을 더해도 배열의 크기만큼 주소값이 증가한다.

## 3. Lab3 ap2-1.c

```
#include <stdio.h>
#define MAX_SIZE 100

float sum1(float list[], int); // sum1 함수 호출
float sum2(float *list, int); // sum2 함수 호출
float sum3(int n, float *list); // sum3 함수 호출

float input[MAX_SIZE], answer; // 변수 선언
int i; // 변수 선언
int main(void){
```

```

for(i=0; i< MAX_SIZE; i++)
    input[i] = i; // input[i] 배열에 i 값 할당
printf("----- 오재식 2022040014 -----");
printf("-----\n");
printf("sum1(input, MAX_SIZE) \n");
printf("-----\n");
printf("input \t = %p\n", input); // input 의 주소값 출력

answer = sum1(input, MAX_SIZE); // answer 에 sum1 return 값 할당
printf("The sum is : %f\n\n", answer); // answer 값 출력

printf("-----\n");
printf("sum2(input, MAX_SIZE) \n");
printf("-----\n");
printf("input \t = %p\n", input); // input 의 주소값 출력

answer = sum2(input, MAX_SIZE); // answer 에 sum2 return 값 할당
printf("The sum is : %f\n\n", answer); // answer 값 출력

printf("-----\n");
printf("sum3(MAX_SIZE, input) \n");
printf("-----\n");
printf("input \t = %p\n", input); // input 의 주소값 출력

answer = sum3(MAX_SIZE, input); // answer 에 sum3 return 값 할당
printf("The sum is : %f\n\n", answer); // answer 값 출력
}

float sum1(float list[0], int n){
    printf("list \t= %p\n", list); //list 의 주소값 출력
    printf("&list \t = %p\n", &list); // list 의 주소값 출력

    int i;
    float tempsum = 0;
    for(i=0; i<n; i++)
        tempsum += list[i]; //list 의 모든 값을 합함.

    return tempsum; // tempsum 값 리턴
}

float sum2(float *list, int n){
    printf("list \t= %p\n", list); //list 의 주소값 출력
    printf("&list \t = %p\n", &list); // list 의 주소값 출력

    int i;
    float tempsum = 0;
    for(i=0; i<n; i++)
        tempsum += list[i]; //list 의 모든 값을 합함.

```

```

    return tempsum; // tempsum 값 리턴
}

float sum3(int n, float *list){
    printf("list \t= %p\n", list); //list 의 주소값 출력
    printf("&list \t = %p\n", &list); // list 의 주소값 출력

    int i;
    float tempsum = 0;
    for(i=0; i<n; i++)
        tempsum += list[i]; //list 의 모든 값을 합함.

    return tempsum; // tempsum 값 리턴
}

```

```

----- 오재식 2022040014 -----
sum1(input, MAX_SIZE)
-----
input    = 0x10069c008
list     = 0x10069c008
&list    = 0x16f76b248
The sum is : 4950.000000

-----
sum2(input, MAX_SIZE)
-----
input    = 0x10069c008
list     = 0x10069c008
&list    = 0x16f76b248
The sum is : 4950.000000

-----
sum3(MAX_SIZE, input)
-----
input    = 0x10069c008
list     = 0x10069c008
&list    = 0x16f76b248
The sum is : 4950.000000

```

함수에 마라미터순서대로 메모리에 쌓이기 때문에 파라미터의 순서가 바뀌었을시 할당되는 주소값이 바뀐다.

#### 4. Lab3 ap2-2.c

```

#include <stdio.h>

void print_one(int *ptr, int rows); //print_one 함수 호출

int main(){
    int one[] = {0,1,2,3,4}; // one 배열에 값 할당
    printf("----- 오재식 2022040014 -----");
    printf("one = %p\n", one); // one 의 주소값 출력
    printf("&one = %p\n", &one); // one 의 주소값 출력
    printf("&one[0] = %p\n", &one[0]); // one[0] 번째 주소값 출력
    printf("\n");

    printf("-----\n");
    printf("  printf_one(&one[0], 5) \n");
    printf("-----\n");
    print_one(&one[0], 5); //print_one 함수 실행

    printf("-----\n");
}

```

```

printf(" print_one(one, 5) \n");
printf("-----\n");
print_one(one, 5); // print_one 함수 실행

return 0;
}

void print_one(int *ptr, int rows){
    int i;
    printf("Address \t Contents \n");
    for(i = 0; i < rows; i++)
        printf("%p \t %5d\n", ptr + i, *(ptr+i)); //포인터 ptr로 받아온 값들의 주소값을
rows 까지 출력
    printf("\n");
}

```

```

----- 오재식 2022040014 -----
one = 0x16fa3f2b0
&one = 0x16fa3f2b0
&one[0] = 0x16fa3f2b0

```

```

printf_one(&one[0], 5)

```

Address	Contents
0x16fa3f2b0	0
0x16fa3f2b4	1
0x16fa3f2b8	2
0x16fa3f2bc	3
0x16fa3f2c0	4

```

print_one(one, 5)

```

Address	Contents
0x16fa3f2b0	0
0x16fa3f2b4	1
0x16fa3f2b8	2
0x16fa3f2bc	3
0x16fa3f2c0	4

&one[0]과 one의 주소값이 일치하기 때문에 같은 결과가 출력된다.

## 5. Lab3 size.c

```

#include <stdio.h>
#include <stdlib.h>

int main(){
    int **x;
    printf("----- 오재식 2022040014 -----");
    printf("sizeof(x) = %lu\n", sizeof(x)); //변수 x에 할당된 메모리 사이즈 출력
    printf("sizeof(*x) = %lu\n", sizeof(*x)); // *x에 할당된 메모리 사이즈 출력
    printf("sizeof(**x) = %lu \n", sizeof(**x)); // **x에 할당된 메모리 사이즈 출력

    return 0;
}

```

```

----- 오재식 2022040014 -----
sizeof(x) = 8
sizeof(*x) = 8
sizeof(**x) = 4

```

x는 이중포인터 \*x는 포인터 \*\*x는 int형이기 때문에 각각 8,8,4 바이트의 메모리 값이 할당되었음을 알 수 있다.

## 6. Lab3 struct.c

```
#include <stdio.h>
#include <stdlib.h>
struct student1 { //구조체 선언
    char lastName;
    int studentId;
    char grade;
};

typedef struct { //구조체 선언
    char lastName;
    int studentId;
    char grade;
}student2;

int main(){
    struct student1 st1 = {'A', 100, 'A'}; //student1 형 변수에 선언후 값 할당
    printf("----- 오재식 2022040014 -----\\n");
    printf("st1.lastName = %c\\n", st1.lastName); // st1.lastName 출력
    printf("st1.studentId = %d\\n", st1.studentId); // st1.studentId 출력
    printf("st1.grade = %c\\n", st1.grade); // st1.grade 출력

    student2 *st2 = malloc(sizeof(student2)); //student2 형 포인터 변수 선언후 메모리 할당
    st2->lastName = 'B'; //화살표 함수를 사용하여 값 할당
    st2->studentId = 200;
    st2->grade = 'B';
    printf("st2.lastName = %c\\n", st2->lastName); //포인터 st2 가 가리키는 주소값의
lastName 출력
    printf("st2.studentId = %d\\n", st2->studentId); //포인터 st2 가 가리키는 주소값의
studentId 출력
    printf("st2.grade = %c\\n", st2->grade); //포인터 st2 가 가리키는 주소값의 grade 출력

    student2 *st3; //student2 형 포인터 변수 선언
    st3 = st2; // st3 에 st2 값 할당
    printf("\\nst3.lastName = %c\\n", st3->lastName); //포인터 st3 가 가리키는 주소값의
lastName 출력
    printf("st3.studentId = %d\\n", st3->studentId); //포인터 st3 가 가리키는 주소값의
studentId 출력
    printf("st3.grade = %c\\n", st3->grade); //포인터 st3 가 가리키는 주소값의 grade 출력
    if(st2 == st3) //st2 와 st3 가 동치이면 equal, 아니면 not equal 출력
        printf("equal\\n");
    else
        printf("not equal\\n");

    return 0;
}
```

```

----- 오재식 2022040014 -----
st1.lastName = A
st1.studentId = 100
st1.grade = A
st2.lastName = B
st2.studentId = 200
st2.grade = B

st3.lastName = B
st3.studentId = 200
st3.grade = B
equal

```

구조체를 가리키는 포인터 두개를 만들고 동적할당을 통해 구조체 크기만큼의 메모리를 할당한 후 if문을 사용하니 정상적으로 작동하였다.

## 7. Lab3 padding.c

```

#include <stdio.h>

struct student { //구조체 선언
    char lastName[13];
    int studentId;
    short grade;
};

int main(){
    struct student pst; // student 형 변수 선언
    printf("----- 오재식 2022040014 -----");
    printf("size of student = %ld\n", sizeof(struct student)); //구조체 student 의
    메모리 크기 출력
    printf("size of int = %ld\n", sizeof(int)); //int 의 메모리 크기 출력
    printf("size of short = %ld\n", sizeof(short)); // short 의 메모리 크기 출력

    return 0;
}

```

```

----- 오재식 2022040014
size of student = 24
size of int = 4
size of short = 2

```

구조체 안의 선언된 변수들의 메모리 크기를 다 더한 값보다 구조체의 메모리 크기가 큰 이유는 메모리 할당에서 빈 곳을 채우는 것이 아니라 4,8,16 단위로 메모리를 할당하면서 빈공간이 있다. Char 13bytes -> 16bytes, int 4bytes, short 2bytes -> 4bytes



```

→ 자료구조 git:(main) ✕ git add .
→ 자료구조 git:(main) ✕ git commit -m "자료구조 과제"
[main e1ac201] 자료구조 과제
Committer: 오재식 <ojaesig@ojaesigs-MacBook-Air.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

16 files changed, 170 insertions(+)
create mode 100644 .vscode/tasks.json
create mode 100755 ap1
create mode 100644 ap1.dSYM/Contents/Info.plist
create mode 100644 ap1.dSYM/Contents/Resources/DWARF/ap1
create mode 100755 ap2
create mode 100755 ap2-1
create mode 100755 ap2-2
[ create mode 100644 ap2-2.c
create mode 100644 ap2.dSYM/Contents/Info.plist
create mode 100644 ap2.dSYM/Contents/Resources/DWARF/ap2
create mode 100755 padding
create mode 100644 padding.c
create mode 100755 size
create mode 100644 size.c
create mode 100755 struct
create mode 100644 struct.c
→ 자료구조 git:(main) git push origin main
[warning: redirecting to https://github.com/ohjaesik/homework3.git/
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 8 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (27/27), 11.18 KiB | 3.73 MiB/s, done.
Total 27 (delta 9), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (9/9), done.
To http://github.com/ohjaesik/homework3.git
85e0756..e1ac201  main -> main

```

```

fatal: Authentication failed for 'https://github.com/ohjaesik/homework3
→ 자료구조 git:(main) ✖ git push origin main
Username for 'https://github.com': ohjaesik
[Password for 'https://ohjaesik@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-your-account/managing-your-account/turning-off-password-authentication
fatal: Authentication failed for 'https://github.com/ohjaesik/homework3
→ 자료구조 git:(main) ✖ git push origin main
Username for 'https://github.com': ohjaesik
[Password for 'https://ohjaesik@github.com':
warning: redirecting to https://github.com/ohjaesik/homework3.git/
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 755 bytes | 755.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://github.com/ohjaesik/homework3.git
 * [new branch]      main -> main
→ 자료구조 git:(main) ✖ git add ap2.c
→ 자료구조 git:(main) ✖ git commit -m "ap2.c"
[main f348ddc] ap2.c
Committer: 오재식 <ojaesig@ojaesigs-MacBook-Air.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 25 insertions(+)
create mode 100644 ap2.c
→ 자료구조 git:(main) ✖ git push origin main
warning: redirecting to https://github.com/ohjaesik/homework3.git/
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 687 bytes | 687.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To http://github.com/ohjaesik/homework3.git
 939d6ce..f348ddc main -> main
→ 자료구조 git:(main) ✖ git add ap2-1.c
→ 자료구조 git:(main) ✖ git commit -m "ap2-1.c"
[main 85e0756] ap2-1.c
Committer: 오재식 <ojaesig@ojaesigs-MacBook-Air.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 74 insertions(+)
create mode 100644 ap2-1.c

```

<https://github.com/ohjaesik/homework3>